Petascale Computing for Large-Scale Graph Problems

Conference Paper · April 2008

DOI: 10.1109/CISIS.2008.150 · Source: IEEE Xplore

David Bader
New Jersey Institute of Technology

Petascale Computing for Large-Scale Graph Problems

David A. Bader*

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332 USA

Abstract. Graph theoretic problems are representative of fundamental kernels in traditional and emerging computational sciences such as chemistry, biology, and medicine, as well as applications in national security. Yet they pose serious challenges for parallel machines due to non-contiguous, concurrent accesses to global data structures with low degrees of locality. Few parallel graph algorithms outperform their best sequential implementation due to long memory latencies and high synchronization costs. In this talk, we consider several graph theoretic kernels for connectivity and centrality and discuss how the features of petascale architectures will affect algorithm development, ease of programming, performance, and scalability.

1 Petascale Computing

Computational science enables us to investigate phenomena where economics or constraints preclude experimentation, evaluate complex models and manage massive data volumes, model processes across interdisciplinary boundaries, and transform business and engineering practices. Increasingly, cyberinfrastructure is required to address our national and global priorities, such as sustainability of our natural environment by reducing our carbon footprint and by decreasing our dependencies on fossil fuels, improving human health and living conditions, understanding the mechanisms of life from molecules and systems to organisms and populations, preventing the spread of disease, predicting and tracking severe weather, recovering from natural and human-caused disasters, maintaining national security, and mastering nanotechnologies. Several of our most fundamental intellectual questions also require computation, such as the formation of the universe, the evolution of life, and the properties of matter.

Realizing that cyberinfrastructure is essential to research innovation and competitiveness, several nations are now in a “new arms race to build the world’s mightiest computer” (John Markoff, *New York Times*, August 19, 2005). These petascale computers, expected around 2008 to 2012, will perform 10^{15} operations per second, nearly an order of magnitude faster than today’s speediest

---

* This keynote talk presented joint work with Kamesh Madduri. This work was supported in part by NSF Grants CNS-0614915, CAREER CCF-0611589, and DBI-0420513; and DARPA contract NBCH30390004.

R. Wyrzykowski et al. (Eds.): PPAM 2007, LNCS 4967, pp. 166–169, 2008.

supercomputer. In fact several nations are in a worldwide race to deliver high-performance computing systems that can achieve 10 petaflops or more within the next five years.

2 Massive Graph Theoretic Applications

The modeling and analysis of massive, dynamically evolving semantic networks raises new and challenging research problems to respond quickly to complex queries. Empirical studies on real-world systems such as the Internet, socio-economic interactions, and biological networks have revealed that they exhibit common structural features – a low graph diameter, skewed vertex degree distribution, self-similarity, and dense sub-graphs. Analogous to the small-world (short paths) phenomenon, these real-world data sets are broadly referred to and modeled as *small-world* networks [1,2]. Our research highlights the design and implementation of novel high performance computing approaches to efficiently solve advanced small-world network analysis queries, enabling analysis of networks that were previously considered too large to be feasible.

For tractable analysis of large-scale networks, we present SNAP(Small-world Network Analysis and Partitioning) [3], an open-source graph analysis framework. SNAPis a modular infrastructure that provides an optimized collection of *algorithmic building blocks* (efficient implementations of key graph-theoretic analytic approaches) to end-users. In prior work, we have designed novel parallel algorithms for several graph problems that run efficiently on shared memory systems. Our implementations of breadth-first graph traversal [4], shortest paths [5,6], spanning tree [7], minimum spanning tree, connected components [8], and other problems achieve impressive parallel speedup for arbitrary, sparse graph instances. We redesign and integrate several of our recent parallel graph algorithms into SNAP, with additional optimizations for social networks. Thus, SNAPprovides a simple and intuitive interface for the network analyst, effectively hiding the parallel programming complexity involved in low-level algorithm design from the user while providing a productive high-performance environment for complex queries.

2.1 Centrality Analysis Queries

One of the fundamental problems in network analysis is to determine the *importance* (or the *centrality*) of a given entity in a network. Some of the well-known metrics for computing centrality are closeness, stress and betweenness [9]. Of these indices, betweenness has been extensively used in recent years for the analysis of social-interaction networks, as well as other large-scale complex networks. Some applications include lethality in biological networks, study of sexual networks and AIDS, organizational behavior, supply chain management processes, as well as identifying key actors in terrorist networks. Betweenness is also used as the primary routine in accepted social network analysis algorithms for clustering and community identification in real-world networks.

Betweenness is a global centrality metric that is based on shortest-path enumeration. It is compute-intensive with a quadratic time complexity in the number of vertices. We explore high performance computing techniques [10] that exploit the typical small-world graph topology to speed up exact, as well as approximate, centrality computation. We demonstrate the capability to compute betweenness on networks that are three orders of magnitude larger than ones that can be processed by state-of-the-art network analysis packages. In contrast to existing approaches, we use a global topological measure for centrality queries in a large network and also support this metric in SNAP.

2.2 Path-Based Queries

Several common analysis queries can be naturally formulated as path-based problems. For instance, while analyzing a collaboration network, we might be interested in chains of publications or patents relating the work of two researchers. In a social network, relationship attributes are frequently encapsulated in the edge type, and we may want to discover paths formed by the composition of specific relationships (for instance, *subordinate of* and *friend of*). SNAPsupports common variants of shortest-path and flow-based query formulations. Other related advanced queries include *subgraph isomorphism* (finding an exact or approximate pattern in the large graph) and *connection subgraphs* (informally, finding a subgraph relating two entities of interest) that are extensions of simpler path-based algorithms.

2.3 Automated Community Detection

A key problem in social network analysis is that of finding communities, dense components, or detecting other latent structure. In recent work, we designed three new clustering schemes (two hierarchical agglomerative approaches, and one divisive clustering algorithm) [3] that optimize modularity, a popular clustering measure. We also conducted an extensive experimental study and demonstrated that our parallel schemes give significant running time improvements over existing modularity-based clustering heuristics. For instance, our novel divisive clustering approach based on approximate edge betweenness centrality is **more than two orders of magnitude faster** than the Newman-Girvan algorithm [11] on a multicore computer, while maintaining comparable clustering quality.

3 Summary

The analysis of massive graphs requires petascale computing systems. We discuss the design and implementation of efficient parallel algorithms for novel community structure identification, classical graph-theoretic kernels, topological indices that provide insight into the network structure, and preprocessing kernels for small-world graphs. Our results demonstrate that these parallel approaches are several orders of magnitude faster than competing algorithms – this enables analysis of networks that were previously considered too large to be tractable. As part of ongoing work, we are designing new small-world network analysis kernels and incorporating existing techniques into SNAP. Our current focus is on the design of novel algorithms for massive-scale small-world networks, including better techniques for centrality analysis, path-based queries, spectral partitioning, and community detection.

References

1. Watts, D., Strogatz, S.: Collective dynamics of small world networks. Nature 393, 440–442 (1998)
2. Amaral, L., Scala, A., Barthélémy, M., Stanley, H.: Classes of small-world networks. Proceedings of the National Academy of Sciences USA 97, 11149–11152 (2000)
3. Bader, D., Madduri, K.: SNAP, Small-world Network Analysis and Partitioning: an open-source parallel graph framework for the exploration of large-scale networks. In: Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS 2008), Miami, FL (2008)
4. Bader, D., Madduri, K.: Designing multithreaded algorithms for breadth-first search and st-connectivity on the Cray MTA-2. In: Proc. 35th Int'l Conf. on Parallel Processing (ICPP), Columbus, OH, IEEE Computer Society, Los Alamitos (2006)
5. Madduri, K., Bader, D., Berry, J., Crobak, J.: An experimental study of a parallel shortest path algorithm for solving large-scale graph instances. In: Proc. The 9th Workshop on Algorithm Engineering and Experiments (ALENEX 2007), New Orleans, LA (2007)
6. Crobak, J., Berry, J., Madduri, K., Bader, D.: Advanced shortest path algorithms on a massively-multithreaded architecture. In: Proc. Workshop on Multithreaded Architectures and Applications, Long Beach, CA (2007)
7. Bader, D.A., Cong, G.: A fast, parallel spanning tree algorithm for symmetric multiprocessors (SMPs). Journal of Parallel and Distributed Computing 65, 994–1006 (2005)
8. Bader, D., Cong, G., Feo, J.: On the architectural requirements for efficient execution of graph algorithms. In: Proc. 34th Int'l Conf. on Parallel Processing (ICPP), Oslo, Norway, IEEE Computer Society, Los Alamitos (2005)
9. Freeman, L.: A set of measures of centrality based on betweenness. Sociometry 40, 35–41 (1977)
10. Bader, D., Madduri, K.: Parallel algorithms for evaluating centrality indices in real-world networks. In: Proc. 35th Int'l Conf. on Parallel Processing (ICPP), Columbus, OH, IEEE Computer Society, Columbus, OH, IEEE Computer Society, Los Alamitos (2006)
11. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences USA 99, 7821–7826 (2002)